
Subject: Witz and map variable

Posted by [Zbych](#) on Tue, 27 Nov 2012 07:51:06 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I would like to replace some numbers in witz script with string.

Do something similar to `Format("%[0:OK;1:ERROR;???]s", error_code)`.

I guess that map variable should do the trick. Can someone show me example how to create and use map variable in witz?

Subject: Re: Witz and map variable

Posted by [dolik.rce](#) on Tue, 27 Nov 2012 08:45:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Zbych wrote on Tue, 27 November 2012 08:51Hi,

I would like to replace some numbers in witz script with string.

Do something similar to `Format("%[0:OK;1:ERROR;???]s", error_code)`.

I guess that map variable should do the trick. Can someone show me example how to create and use map variable in witz?

Hi Zbych,

Using maps in witz is quite straightforward. You just pass it to Skylark in handler:SKYLARK(MyHandler,"path"){

```
    int i=1;
    ValueMap map; // ValueArray could be used as well
    map.Add(0,"OK");
    map.Add(1,"ERROR");
    http("ID",i)("MAP",map).RenderResult("MyApp/template");
}
```

and then you can work with it in the template:element \$ID in MAP is: \$MAP[ID]

I haven't tested the code, hopefully I made no mistakes

Best regards,

Honza

Subject: Re: Witz and map variable

Posted by [Zbych](#) on Tue, 27 Nov 2012 09:50:10 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks for reply.

I know that I can pass the map from handler. The problem is that this map will be different for every language and I would like to keep all language things in one place (witz scripts).

That is why I am looking for something like this:

```
error_map = { 0:"OK", 1:"ERROR", 2:"???" }  
<b> System status: $error_map[$error_code] </b>
```

Is it possible to create new variables inside witz scripts?

Subject: Re: Witz and map variable
Posted by [dolik.rce](#) on Tue, 27 Nov 2012 11:29:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Zbych wrote on Tue, 27 November 2012 10:50 Thanks for reply.
I know that I can pass the map from handler. The problem is that this map will be different for every language and I would like to keep all language things in one place (witz scripts).

Ok, I think I understand now. I think your best chance if you really want to keep it that way is to create a witz function, something like `translate(lang,word1,word2,word3,...)` that would just return the parameter according to the value of lang. But I think this is not really the correct way, as it might require a lot of work when adding support for new language. You would have to modify all of your templates to add another parameter, even if the same word would be used repeatedly...

So what about a function that would utilize U++ internationalization capabilities? It could behave exactly as `t_()` macro, then you could just write `t_("String in English")` and it would return the correct translation based on .t files. I know it violates your wish to "have everything language specific in witz", but .t files are designed for this, so it might not be so bad idea

I assume you are using the `.lang/.__lang__` variables and language specific templates (as described in chapter 11 of the tutorial), right?

Zbych wrote on Tue, 27 November 2012 10:50 Is it possible to create new variables inside witz scripts?

No, quoting from the manual:Quote:Witz primary values come from shared variable space of Skylark handler; the only other way to define a new variable is by using `for` statements.

Honza

Subject: Re: Witz and map variable
Posted by [Zbych](#) on Wed, 28 Nov 2012 18:50:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thank you Honza for clarification. I guess that creation of variables in witz should be added to the

next wish list

Subject: Re: Witz and map variable

Posted by [dolik.rce](#) on Thu, 29 Nov 2012 06:03:25 GMT

[View Forum Message](#) <> [Reply to Message](#)

Zbych wrote on Wed, 28 November 2012 19:50 Thank you Honza for clarification. I guess that creation of variables in witz should be added to the next wish list

Well, it is kind of a philosophic question... I think that at the moment you put custom variables to templates, the designer might as well request more and more "reasonable" features to work with them and you end up with half the codebase located in templates. Which is of course less effective, than having it in pure C++.

If there is a need for something like this, it should be IMHO handled through a function and only used in some special cases, not as a general rule.

Honza

Subject: Re: Witz and map variable

Posted by [Zbych](#) on Thu, 29 Nov 2012 09:37:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Perhaps you are right. But know I need to keep some language specific things in *.t files and some in *.witz.

I have one more question for you.

I have a drop list and want to preselect one value from it.

Right now I create a map ("Item1", "0")("Item2", "1")("Item3", "0"), pass it to witz and use \$if to preselect position.

```
<select name="MyList">
$for(i in MYLIST)
  $if ( i != 0 )
    <option value="$i._key" selected="selected">$i._key</option>
  $else
    <option value="$i._key">$i._key</option>
$/
$/
</select>
```

Is there a shorter way (without \$if)? Maybe something similar to c?

```
$for(i in MYLIST)
<option value="$i._key" $(i != 0) ? selected="selected" : "">$i._key</option>
$/  
$/  

```

Subject: Re: Witz and map variable
Posted by [dolik.rce](#) on Thu, 29 Nov 2012 10:11:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Zbych wrote on Thu, 29 November 2012 10:37 I have one more question for you.
I have a drop list and want to preselect one value from it.
Right now I create a map ("Item1", "0")("Item2", "1")("Item3", "0"), pass it to witz and use \$if to preselect position.

```
<select name="MyList">
$for(i in MYLIST)
$if ( i != 0 )
<option value="$i._key" selected="selected">$i._key</option>
$else
<option value="$i._key">$i._key</option>
$/  
$/  

```

</select> I'm not aware of any other simple way, I do it like this too. Only alternative I know about is sending MYLIST as an array of arrays/maps of two field, where the second contains the "selected=\"selected\" or empty string, so it can be included in single pass, without \$if. But it seems to me even worse, because it pushes html code into app

Zbych wrote on Thu, 29 November 2012 10:37 Is there a shorter way (without \$if)? Maybe something similar to c?

```
$for(i in MYLIST)
<option value="$i._key" $(i != 0) ? selected="selected" : "">$i._key</option>
$/ I believe (can't test right now) you can inline the $if:$for(i in MYLIST)
<option value="$i._key" $if(i != 0) selected="selected" $/>$i._key</option>
$/  

```

Honza

Subject: Re: Witz and map variable
Posted by [Zbych](#) on Thu, 29 Nov 2012 10:26:32 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Thu, 29 November 2012 11:11 I believe (can't test right now) you can inline the \$if:\$for(i in MYLIST)
<option value="\$i._key" \$if(i != 0) selected="selected" \$/>\$i._key</option>
\$/

This is it! Thank you.

I have one more problem with map. I can not display map with key zero:

```
*.cpp
ValueMap yn;
yn.Add(0, "No");
yn.Add(1, "Yes");
http
("YESNO", yn)
[...]
```

```
*.witz:
$YESNO[0] <-- displays nothing
$YESNO[1]
```

I am quite sure I already saw this in forum, but I can not find it now.

Subject: Re: Witz and map variable
Posted by [dolik.rce](#) on Thu, 29 Nov 2012 13:55:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

Zbych wrote on Thu, 29 November 2012 11:26I have one more problem with map. I can not display map with key zero:

```
*.cpp
ValueMap yn;
yn.Add(0, "No");
yn.Add(1, "Yes");
http
("YESNO", yn)
[...]
```

```
*.witz:
$YESNO[0] <-- displays nothing
$YESNO[1]
```

I am quite sure I already saw this in forum, but I can not find it now.

I think you mean this post. That problem was fixed, but it seems it only works for ValueArray. Try using that or ValueMap with some other indices...

Honza

Subject: Re: Witz and map variable
Posted by [Zbych](#) on Thu, 29 Nov 2012 20:18:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

It works with ValueArray. Thanks.

Subject: Re: Witz and map variable
Posted by [dolik.rce](#) on Fri, 30 Nov 2012 06:41:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Ha, I finally found the actual problem!

It all goes down to the compiler, it decides that the 0 is not integer, but empty string on the line
yn.Add(0,"NO") So it saves it to the map with "" as key, not 0. To make this work, you have to
cast the 0 to Value first: ValueMap yn;
yn.Add(Value(0), "YES");
yn.Add(1,"NO");

Best regards,
Honza

Subject: Re: Witz and map variable
Posted by [Zbych](#) on Fri, 30 Nov 2012 06:51:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Fri, 30 November 2012 07:41
It all goes down to the compiler, it decides that the 0 is not integer, but empty string

Bug or feature?

Subject: Re: Witz and map variable
Posted by [dolik.rce](#) on Fri, 30 Nov 2012 07:30:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Zbych wrote on Fri, 30 November 2012 07:51dolik.rce wrote on Fri, 30 November 2012 07:41
It all goes down to the compiler, it decides that the 0 is not integer, but empty string

Bug or feature?

I'd say it is not a feature Perhaps a limitation

I think it is possible to solve by adding new overload of Add (and perhaps also similar one for Set):
`void Add(int n, const Value& value) { Add(Value(n), value); }`

Mirek, what do you think? I'm not familiar enough with the Value * internals, so I can't really tell if it breaks anything else

Honza

Subject: Re: Witz and map variable

Posted by [mirek](#) on Sat, 01 Dec 2012 11:17:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

dolik.rce wrote on Fri, 30 November 2012 02:30

I think it is possible to solve by adding new overload of Add (and perhaps also similar one for Set):
`void Add(int n, const Value& value) { Add(Value(n), value); }`

Mirek, what do you think? I'm not familiar enough with the Value * internals, so I can't really tell if it breaks anything else

Honza

Agreed. It is not the most elegant solution, but if we want to be able to use ValueMap directly with Id and SqlId keys, it is probably the only possible solution...

(fixed)

Mirek
